Notes for “LP1” and “Structure and matrix sparsity”

Overview

“LP1” and “Structure and matrix sparsity” are three linked hours of the NATCOR convex optimization course. The “LP1” lecture will present the theoretical foundations of linear programming (LP) and the simplex method for solving LP problems.

Although theory and computational techniques for the simplex and interior point methods for solving LP problems are well-established, the challenge of solving large scale problems and exploiting high performance computing environments remain important research goals. The “Structure and matrix sparsity” lecture is part of the “Theoretical” stream and will focus on how the structure and matrix sparsity of LP problems may be exploited to yield efficient solution techniques.

“LP1”

Introduction

The general LP problem is

$$\text{maximize } f = \boldsymbol{c}^T\boldsymbol{x} \quad \text{subject to } A\boldsymbol{x} \leq \boldsymbol{b}, \quad \boldsymbol{x} \geq \boldsymbol{0} \tag{1}$$

The problem has n **variables** and m **constraints**. It is the fundamental model in optimal decision-making and solutions of small examples via the **simplex algorithm** can be obtained relatively easily so both are generally studied as part of introductory Operational Research or optimization courses.

Many students attending “LP1” will be familiar with the theoretical foundations of LP, the simplex algorithm and its implementation via the tableau simplex method so LP1 will be largely revision. The purpose of the background reading is to present the material of LP1 in note form, so anyone attending who is unfamiliar with it before should study this document carefully. Although the presentation is mathematical, intuitive interpretation is also given with the aim of “LP1” being accessible to those who do not have a Mathematics degree.

General LP theory

The set of points satisfying the inequalities and bounds of (1) is referred to as the **feasible region**. Mathematically, it is **convex polyhedron** in $\mathbb{R}^n$ whose **vertices** are at the intersection of n constraint/bound planes. If the feasible region is empty then the LP is said to be **infeasible**. Otherwise, either the objective is unbounded on the feasible region [and the LP is said to be **unbounded**] or the LP has an **optimal solution**.

Vertex characterization and algebraic analysis are aided by introducing a **slack variable** x_{n+i} for each inequality $a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i$, transforming it into an equation and bound on x_{n+i}

$$a_{i1}x_1 + \ldots + a_{in}x_n + x_{n+i} = b_i \quad x_{n+i} \geq 0$$

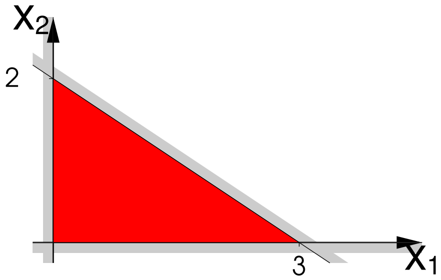
After adding slack variables for all constraints the problem is said to be in **standard form**

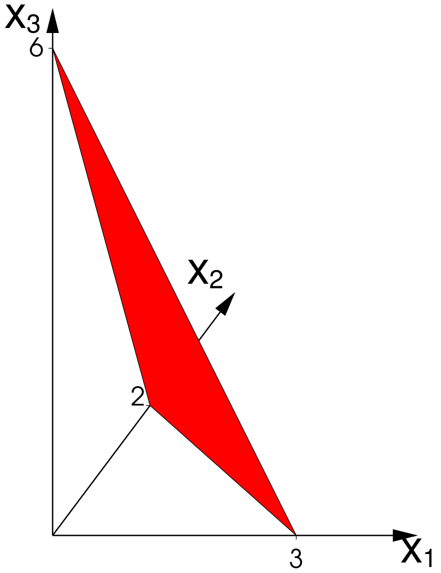
$$\text{maximize } f = \overline{\boldsymbol{c}}^T\boldsymbol{x} \quad \text{subject to } \overline{A}\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{x} \geq \boldsymbol{0} \tag{2}$$

where $\overline{A} = \begin{bmatrix} A & I \end{bmatrix}$ and $\overline{\boldsymbol{c}} = \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{0} \end{bmatrix}$.

Example:
For the LP

$$\begin{array}{llcl}
\max & f & = & 3x_1 \;+\; 5x_2 \\
\text{s. t.} & & & 2x_1 \;+\; 3x_2 \;\leq\; 6 \\
 & & & x_1 \geq 0 \quad \text{and} \quad x_2 \geq 0
\end{array}$$



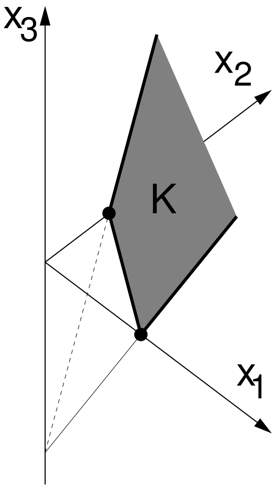


Add slack variable x_3 to give

$$\begin{array}{llcl}
\max & f & = & 3x_1 \;+\; 5x_2 \\
\text{s. t.} & & & 2x_1 \;+\; 3x_2 \;+\; x_3 \;=\; 6 \\
 & & & x_1 \geq 0, \quad x_2 \geq 0 \quad \text{and} \quad x_3 \geq 0
\end{array}$$

- Vertices correspond to choosing 2 of 3 variables to be zero
- Use the equation to solve for the value of the third variable

Mathematically, the system of equations in (2) has a particular solution $\boldsymbol{x}_b = \begin{bmatrix}\mathbf{0}\\ \boldsymbol{b}\end{bmatrix}$ and, since $\overline{A}$ has rank m, the vectors $\boldsymbol{v}$ such that $\overline{A}\boldsymbol{v} = \mathbf{0}$ form a space V of dimension n. Hence the solutions of $\overline{A}\boldsymbol{x} = \boldsymbol{b}$ are given by $\boldsymbol{x} = \boldsymbol{x}_b + \boldsymbol{v}$ for all $\boldsymbol{v} \in V$. This is an n-dimensional hyper-plane V_b in $\mathbb{R}^{n+m}$. The **feasible region** K is the intersection of V_b and the orthant $\boldsymbol{x} \geq \mathbf{0}$. It is a convex set for which a vertex is a point $\boldsymbol{x} \in K$ which does not lie strictly within any line segment joining two points in K. It may be shown that if an LP has an optimal solution then there is an optimal solution at a vertex



The **simplex algorithm** for solving LP problems is built on the concept of a **basic solution** of an LP problem:

Definition: The point $\boldsymbol{x} \in \mathbb{R}^{n+m}$ is a **basic solution** of an LP problem in standard form if there is a **partition** of $\{1, 2, \ldots, n+m\}$ into

- A set $\mathcal{N}$ of n indices of **nonbasic variables** with value zero at $\boldsymbol{x}$
- A set $\mathcal{B}$ of m indices of **basic variables** whose values are then uniquely defined by the m equations

Corresponding to the partition of $\{1, 2, \ldots, n+m\}$ into $\mathcal{B}$ and $\mathcal{N}$ the following are defined:

- Corresponding to the n indices in $\mathcal{N}$
  - The matrix $N \in \mathbb{R}^{m \times n}$ formed from the n columns of $\overline{A}$
  - The vector $\boldsymbol{x}_N$ of **nonbasic variables** formed from the n components of $\boldsymbol{x}$
  - The vector $\boldsymbol{c}_N$ of **nonbasic costs** formed from the n components of $\boldsymbol{c}$

- Corresponding to the m indices in $\mathcal{B}$
  - The **basis matrix** $B \in \mathbb{R}^{m\times m}$ formed from the m columns of $\overline{A}$ is nonsingular
  - The vector $\boldsymbol{x}_B$ of **basic variables** formed from the m components of $\boldsymbol{x}$
  - The vector $\boldsymbol{c}_B$ of **basic costs** formed from the m components of $\boldsymbol{c}$

Corresponding to particular sets $\mathcal{B}$ and $\mathcal{N}$, this notation yields the following useful equivalent expression for the LP in standard form (2)

$$\begin{array}{lcccccl} \text{maximize} & f & = & \boldsymbol{c}_B^T\boldsymbol{x}_B & + & \boldsymbol{c}_N^T\boldsymbol{x}_N & \\ \text{subject to} & & & B\boldsymbol{x}_B & + & N\boldsymbol{x}_N & = \ \boldsymbol{b} \\ & & & \multicolumn{3}{l}{\boldsymbol{x}_B \geq \boldsymbol{0},\ \boldsymbol{x}_N \geq \boldsymbol{0}} & \end{array} \tag{3}$$

This is referred to as the **partitioned LP** and it follows from the equations in (3) that since $\boldsymbol{x}_N = \boldsymbol{0}$ at the basic solution $\boldsymbol{x}$, the values of the basic variables at $\boldsymbol{x}$ are given by $\boldsymbol{x}_B = B^{-1}\boldsymbol{b} = \widehat{\boldsymbol{b}}$. If $\widehat{\boldsymbol{b}} \geq \boldsymbol{0}$ then $\boldsymbol{x} \in K$ and is referred to as a **basic feasible solution**. It may be shown that $\boldsymbol{x}$ is a vertex of K iff $\boldsymbol{x}$ is a basic feasible solution.
Thus, since there is an optimal solution of the LP at a vertex (if the LP has an optimal solution) and basic feasible solutions and vertices are equivalent, solution methods need only consider basic feasible solutions.

The following optimality condition for LP problems concludes the LP theory for “LP1”.

Theorem: A sufficient optimality condition for LP problems
A point $\boldsymbol{x} \in K$ is an optimal solution of an LP problem if it is a basic feasible solution with non-positive **reduced costs** $\widehat{\boldsymbol{c}}_N = \boldsymbol{c}_N - N^TB^{-T}\boldsymbol{c}_B$
Proof:

- Relative to $\boldsymbol{x} \in K$, any other point $\boldsymbol{x}' \in K$ is given by $\boldsymbol{x} + \boldsymbol{d}$, where $\boldsymbol{d} = \boldsymbol{x}' - \boldsymbol{x}$

- Let $\boldsymbol{d}$ be partitioned as $\begin{bmatrix}\boldsymbol{d}_B\\ \boldsymbol{d}_N\end{bmatrix}$ and let $\boldsymbol{x}'$ be partitioned as $\begin{bmatrix}\widehat{\boldsymbol{b}}\\ \boldsymbol{0}\end{bmatrix} + \begin{bmatrix}\boldsymbol{d}_B\\ \boldsymbol{d}_N\end{bmatrix}$

- Then, at $\boldsymbol{x}'$, since $B\boldsymbol{d}_B + N\boldsymbol{d}_N = \boldsymbol{0}$ and $\boldsymbol{d}_N \geq \boldsymbol{0}$ follow from the feasibility of $\boldsymbol{x}$ and $\boldsymbol{x}'$,

  $$f = \boldsymbol{c}_B^T(\widehat{\boldsymbol{b}} + \boldsymbol{d}_B) + \boldsymbol{c}_N^T\boldsymbol{d}_N = \widehat{f} - \boldsymbol{c}_B^TB^{-1}N\boldsymbol{d}_N + \boldsymbol{c}_N^T\boldsymbol{d}_N = \widehat{f} + (\boldsymbol{c}_N - N^TB^{-T}\boldsymbol{c}_B)^T\boldsymbol{d}_N = \widehat{f} + \widehat{\boldsymbol{c}}_N^T\boldsymbol{d}_N,$$

  where $\widehat{f} = \boldsymbol{c}_B^T\widehat{\boldsymbol{b}}$ is the objective value at $\boldsymbol{x}$.

- Hence $\widehat{\boldsymbol{c}}_N \leq \boldsymbol{0} \ \Rightarrow\ \widehat{\boldsymbol{c}}_N^T\boldsymbol{d}_N \leq 0 \ \Rightarrow\ f \leq \widehat{f}$ so $\boldsymbol{x}$ is optimal

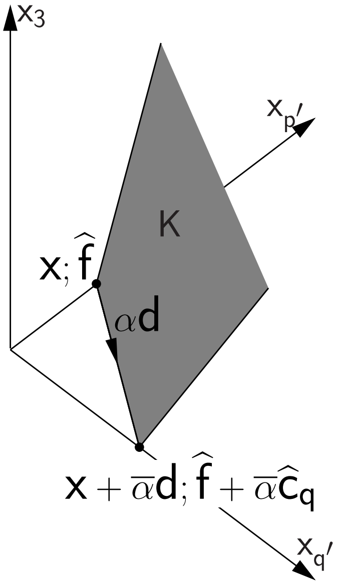
Note:

- This condition $\widehat{\boldsymbol{c}}_N \leq \boldsymbol{0}$ allows the optimality of a basic feasible solution to be checked

- If the condition does not hold, it is used by the **simplex algorithm** to identify a basic feasible solution with better objective value

This is the key to solving LP problems

The simplex algorithm


x_3
$x_{p'}$
K
$x; \widehat{f}$
αd
$x + \overline{\alpha}d; \widehat{f} + \overline{\alpha}\widehat{c}_q$
$x_{q'}$

The simplex algorithm derives from the observation that if, at a basic feasible solution $\boldsymbol{x}$ with $\mathcal{B}$ and $\mathcal{N}$, the optimality condition $\widehat{\boldsymbol{c}}_N \leq \mathbf{0}$ does not hold then there exists an index q such that $\widehat{c}_q > 0$.
If $\boldsymbol{x}$ is partitioned as $\begin{bmatrix}\boldsymbol{x}_B\\ \boldsymbol{x}_N\end{bmatrix}$ then the corresponding step which increases only $x_{q'}$ from zero is given by $\begin{bmatrix}\boldsymbol{d}_B\\ \boldsymbol{d}_N\end{bmatrix}$ for $\boldsymbol{d}_N = \boldsymbol{e}_q$, where $\boldsymbol{e}_q$ is column q of I. Feasibility with respect to the equations of the LP requires $B\boldsymbol{d}_B + N\boldsymbol{d}_N = \mathbf{0}$ so $\boldsymbol{d}_B = -B^{-1}N\boldsymbol{e}_q = -\hat{\boldsymbol{a}}_q$ where $B\hat{\boldsymbol{a}}_q = \boldsymbol{a}_q$ and $\boldsymbol{a}_q$ is column q of N.
If the corresponding nonbasic variable $x_{q'}$ may be increased from zero along $\boldsymbol{x} + \alpha\boldsymbol{d}$ for $\alpha > 0$ whilst maintaining feasibility, then there will be a strict increase $(\alpha\widehat{c}_q)$ in the objective.
For points along $\boldsymbol{x} + \alpha\boldsymbol{d}$ when α is non-negative, the components of $\boldsymbol{x}_N$ remain feasible since $\boldsymbol{d}_N = \boldsymbol{e}_q$.
Thus any limit on the feasibility of $\boldsymbol{x} + \alpha\boldsymbol{d}$ is given by the values $\boldsymbol{x}_B = \widehat{\boldsymbol{b}} - \alpha\hat{\boldsymbol{a}}_q$ of the basic variables. If $\hat{\boldsymbol{a}}_q$ has positive components then, for α sufficiently large, at least one component of $\boldsymbol{x}_B$ will be zeroed. The smallest of these values of α is the greatest step $\overline{\alpha}$ which can be made in the direction $\boldsymbol{d}$ whilst maintaining feasibility. At $\boldsymbol{x} + \overline{\alpha}\boldsymbol{d}$ there is a partition given by the sets $\mathcal{B}$ and $\mathcal{N}$ at $\boldsymbol{x}$ with the index p' of the zeroed basic variable interchanged with index q' of the chosen nonbasic variable. It may be shown that the point $\boldsymbol{x} + \overline{\alpha}\boldsymbol{d}$ is a basic feasible solution corresponding to the new partition.

The above motivational definition of the simplex algorithm is formalised in the following description of its steps

Description of the simplex algorithm: Given a basic feasible solution $\boldsymbol{x}$

1. If the reduced costs are non-positive then **stop: The solution is optimal**
2. Determine the nonbasic variable $x_{q'}$ with most positive reduced cost
3. Determine the feasible direction $\boldsymbol{d}$ when $x_{q'}$ is increased from zero
4. If no basic variable is zeroed on $\boldsymbol{x} + \alpha\boldsymbol{d}$ then **stop: The LP is unbounded**
5. Determine the first basic variable $x_{p'}$ to be zeroed on $\boldsymbol{x} + \alpha\boldsymbol{d}$
6. Make $x_{p'}$ nonbasic and $x_{q'}$ basic
7. Go to 1

Mathematically, the algorithm is defined as follows

Definition of the simplex algorithm: Given a basic feasible solution $\boldsymbol{x}$ with $\mathcal{B}$ and $\mathcal{N}$, iterations of the simplex algorithm are performed as follows:

1. If $\widehat{\boldsymbol{c}}_N \leq \mathbf{0}$ then **stop: the solution is optimal**
2. Determine the index $q' \in \mathcal{N}$ of the variable $x_{q'}$ with most positive reduced cost $\widehat{c}_q$
3. Let $\widehat{\boldsymbol{a}}_q = B^{-1}\boldsymbol{a}_q$, where $\boldsymbol{a}_q$ is column q of N
4. If $\widehat{\boldsymbol{a}}_q \leq \mathbf{0}$ then **stop: the LP is unbounded**

5. Determine the index $p' \in \mathcal{B}$ of the variable $x_{p'}$ corresponding to $p = \operatorname{argmin}^{\substack{i=1,\dots,m \\ \widehat{a}_{iq} > 0}} \dfrac{\widehat{b}_i}{\widehat{a}_{iq}}$
6. Exchange indices p' and q' between $\mathcal{B}$ and $\mathcal{N}$ to yield a new basic feasible solution
7. Go to 1

[Note that index p' is entry p of $\mathcal{B}$ and index q' is entry q of $\mathcal{N}$.]
Each simplex iteration determines an interchange of one index between the sets $\mathcal{B}$ and $\mathcal{N}$. Specifically, the q^{th} entry of $\mathcal{N}$ is interchanged with the p^{th} entry of $\mathcal{B}$. This means that successive basis matrices differ by only one column, and it is essential that this is exploited by implementations of the revised simplex method. Specifically, when the updated basis matrix B' is obtained from B, column p of B is replaced by the vector $\boldsymbol{a}_q$, being column q of N. Since column p of B is the vector $B\boldsymbol{e}_p$ (where $\boldsymbol{e}_p$ is column p of I) and $\boldsymbol{v}\boldsymbol{e}_p^T$ is a zero matrix except for the vector $\boldsymbol{v}$ in column p, B' may be expressed as

$$\begin{aligned} B' &= B + \boldsymbol{a}_q\boldsymbol{e}_p^T - \boldsymbol{a}_p\boldsymbol{e}_p^T \\ &= B\left[I + (\widehat{\boldsymbol{a}}_q - \boldsymbol{e}_p)\boldsymbol{e}_p^T\right] \\ &= BE \quad \text{where} \quad E = I + (\widehat{\boldsymbol{a}}_q - \boldsymbol{e}_p)\boldsymbol{e}_p^T \end{aligned}$$

This result has fundamental theoretical and practical consequences

Theoretically: Since B is nonsingular, B' is nonsingular iff E is nonsingular. This may be established as follows. For a matrix $A = I + \boldsymbol{u}\boldsymbol{v}^T$ the (Sherman-Morrison) formula for A^{-1} is

$$A^{-1} = I - \frac{1}{1 + \boldsymbol{v}^T\boldsymbol{u}}\boldsymbol{u}\boldsymbol{v}^T \quad \text{when} \quad \boldsymbol{v}^T\boldsymbol{u} \neq -1$$

Hence, using $\boldsymbol{u} = \widehat{\boldsymbol{a}}_q - \boldsymbol{e}_p$ and $\boldsymbol{v} = \boldsymbol{e}_p$

$$E^{-1} = I - \frac{1}{\widehat{a}_{pq}}(\widehat{\boldsymbol{a}}_q - \boldsymbol{e}_p)\boldsymbol{e}_p^T$$

Since the pivot value in the simplex iteration is $\widehat{a}_{pq} \neq 0$, it follows that E (and hence B') is nonsingular.

Implementing the simplex algorithm

The data required to perform an iteration of the simplex algorithm consist of

- The **reduced costs** $\widehat{\boldsymbol{c}}_N = \boldsymbol{c}_N - N^T B^{-T}\boldsymbol{c}_B$
- The **pivotal column** $\hat{\boldsymbol{a}}_q = B^{-1}\boldsymbol{a}_q$
- The **reduced RHS** $\widehat{\boldsymbol{b}} = B^{-1}\boldsymbol{b}$

Obtaining these three vectors is the principal computational challenge when solving large-scale LP problems. They may, by inspection, be obtained as follows:

- Solve $B^T\boldsymbol{\pi} = \boldsymbol{c}_B$ and then form the matrix-vector product $N^T\boldsymbol{\pi}$ to obtain $\widehat{\boldsymbol{c}}_N = \boldsymbol{c}_N - N^T\boldsymbol{\pi}$
- Solve $B\hat{\boldsymbol{a}}_q = \boldsymbol{a}_q$ to obtain $\hat{\boldsymbol{a}}_q = B^{-1}\boldsymbol{a}_q$
- Exploit the expression $\boldsymbol{x}_B = \widehat{\boldsymbol{b}} - \overline{\alpha}\hat{\boldsymbol{a}}_q$ for the new values of the basic variables to update $\widehat{\boldsymbol{b}}$

When implemented using these matrix computations, the simplex algorithm is referred to as the **revised simplex method**. Each iteration requires

- The solution of two linear systems with matrix of coefficients B and B^T respectively
- The formation of one matrix-vector product with the matrix N^T

Linear algebra for the revised simplex method

To solve a single linear system in isolation one would consider using Gaussian elimination. However, the computational cost of this is of order m^3 and, in the context of the simplex algorithm, the systems may be solved very much more efficiently. As outlined below, this is possible due to the coefficient matrices being B and B^T and the relation between successive basis matrices B. When this is done the order of the computational cost reduces to m^2.
When Gaussian elimination (GE) is applied to the system $B\boldsymbol{x} = \boldsymbol{b}$ it reduces it to an upper triangular system $U\boldsymbol{x} = \boldsymbol{y}$ which may be solved by back-substitution. In addition, GE yields the **LU decomposition** $B = LU$, where L is a lower triangular matrix with diagonal entries equal to one, and U is an upper triangular matrix. The matrix L consists of the elimination multipliers generated by GE. The LU decomposition may be used to solve any system $B\boldsymbol{x} = \boldsymbol{b}$ as follows.

- Solve the triangular system $L\boldsymbol{y} = \boldsymbol{b}$ by forward substitution
- Solve the triangular system $U\boldsymbol{x} = \boldsymbol{y}$ by backward substitution

Solving triangular system requires one multiplication and addition with each nonzero in the matrix so the computational cost of forward and backward substitution is of order m^2.
Since

$$B' = BE \quad \text{where} \quad E = I + (\widehat{\boldsymbol{a}}_q - \boldsymbol{e}_p)\boldsymbol{e}_p^T \quad \text{and} \quad E^{-1} = I - \frac{1}{\widehat{a}_{pq}}(\widehat{\boldsymbol{a}}_q - \boldsymbol{e}_p)\boldsymbol{e}_p^T$$

$B'\boldsymbol{x} = \boldsymbol{b}$ may be solved as

$$BE\boldsymbol{x} = \boldsymbol{b} \iff B\boldsymbol{y} = \boldsymbol{b}; \qquad \boldsymbol{x} = E^{-1}\boldsymbol{y}$$

so the LU decomposition of B can be re-used, with all costs being of order m^2. Since E has only one non-trivial column, it may be stored using a single vector and the index p. Although this technique can be extended to perform multiple updates, eventually the cost of working with all the matrices of the form E will dominate to the extent that is is preferable to recompute the LU decomposition. This is avoided by an alternative [Fletcher-Matthews] technique which allows the LU decomposition of B' to be obtained by updating the LU decomposition of B at a cost of order m^2, thereby ensuring that all linear systems in the revised simplex method with coefficient matrix B may be solved at a cost of order m^2 each.

Systems of equations with coefficient matrix B^T may be solved as follows using the LU decomposition of B since $B^T = U^T L^T$.

- Solve the lower triangular system $U^T\boldsymbol{y} = \boldsymbol{b}$ by forward substitution
- Solve the upper triangular system $L^T\boldsymbol{x} = \boldsymbol{y}$ by backward substitution

For the updated basis matrix, the system ${B'}^T\boldsymbol{x} = \boldsymbol{b}$ may be solved as

$${B'}^T\boldsymbol{x} = \boldsymbol{b} \iff E^T B^T\boldsymbol{x} = \boldsymbol{b} \iff \boldsymbol{z} = E^{-T}\boldsymbol{b}; \qquad U^T\boldsymbol{y} = \boldsymbol{z}; \qquad L^T\boldsymbol{x} = \boldsymbol{y}$$

The only other significant computational component of the revised simplex method is the matrix-vector product $N^T\boldsymbol{\pi}$ required to obtain $\widehat{\boldsymbol{c}}_N = \boldsymbol{c}_N - N^T\boldsymbol{\pi}$. Given the vector $\boldsymbol{\pi}$, $N^T\boldsymbol{\pi}$ is computed without difficulty since N is readily available as a submatrix of $\overline{A}$.

"Structure and matrix sparsity"

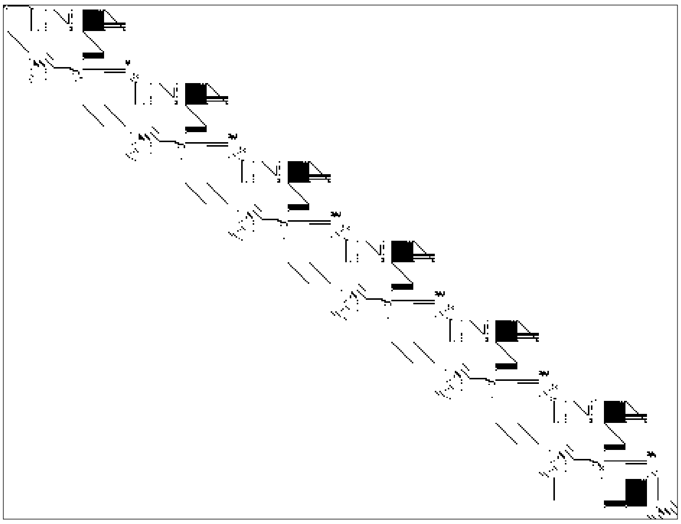
Introduction

Most issues of structure and matrix sparsity in convex optimization can be discussed in relation to the following general LP problem with n variables and m constraints

$$\text{maximize } f = \boldsymbol{c}^T\boldsymbol{x} \quad \text{subject to } A\boldsymbol{x} \le \boldsymbol{b}, \quad \boldsymbol{x} \ge \boldsymbol{0} \tag{4}$$

Such a problem may be modelled directly or formed as a sub-problem when solving discrete or nonlinear problems. By virtue of the modelling process, it is typically the case that most of the entries in the matrix A are zero. The distribution of the nonzeros is generally non-random so the matrix A has **structure**. When it is worth exploiting the existence of these zeros in a computational process the matrix A is said to be **sparse**.

For example, when evaluating a matrix-vector product $\boldsymbol{y} = A\boldsymbol{x}$, each component y_i of the result is the inner product of row i of A and the vector $\boldsymbol{x}$. Thus each entry in A is used exactly once. If most of the entries of A are zero then multiplications and additions with zero will dominate the cost if the computation is implemented simplistically. If these computations with zero are avoided then the calculation becomes much more efficient.



Constraint matrix A for LP test problem STAIR

- $n = 467$ columns (variables)
- $m = 363$ rows (constraints)
- $\tau = 3856$ nonzeros
- $\dfrac{\tau}{n} = \dfrac{3856}{467} = 8.3$ nonzeros/column
- Density $100 \times \dfrac{3856}{467 \times 363} = 2.3\%$

Efficient computations with sparse matrices are achieved in the first instance by storing only the nonzero values in the matrix. This is done row-by-row (or column-by-column), with each nonzero value paired with the index of the column (row) in which it is located. Thus, for example, the inner product between row i of A and the vector $\boldsymbol{x}$ is computed efficiently by multiplying each nonzero in the row by the entry of $\boldsymbol{x}$ corresponding to the column index of the nonzero and summing the results.

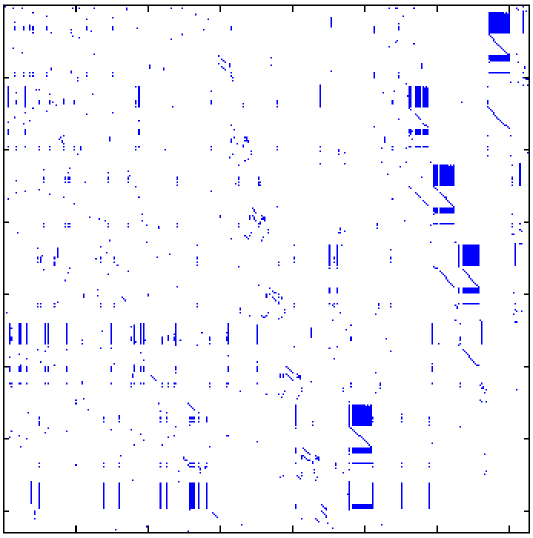
Highly efficient implementations of the revised simplex and interior point methods have been developed and are available as open source or commercial software. These implementations are efficient principally due to exploiting the sparsity of the constraint matrix.

Exploiting sparsity in the revised simplex method

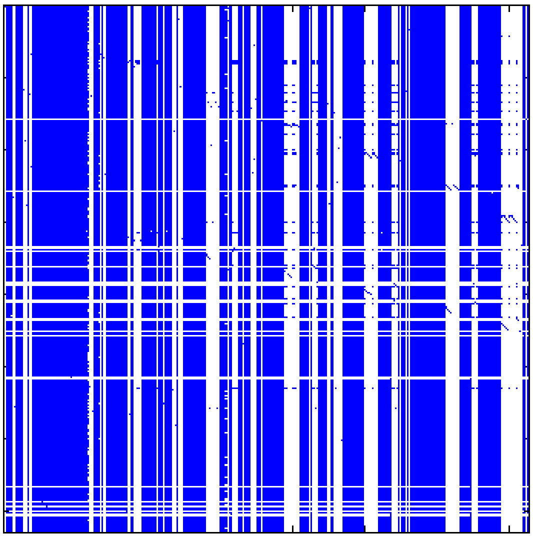
Each iteration of the **revised simplex method** requires

- The solution of two linear systems with matrix of coefficients B and B^T respectively
- The formation of one matrix-vector product with the matrix N^T

In theory: The system $B\boldsymbol{x} = \boldsymbol{b}$ may be solved by computing B^{-1} and forming $\boldsymbol{x} = B^{-1}\boldsymbol{b}$. However, for an LP problem with sparse A, the matrix B is sparse but B^{-1} is typically very dense so computing it is very inefficient.

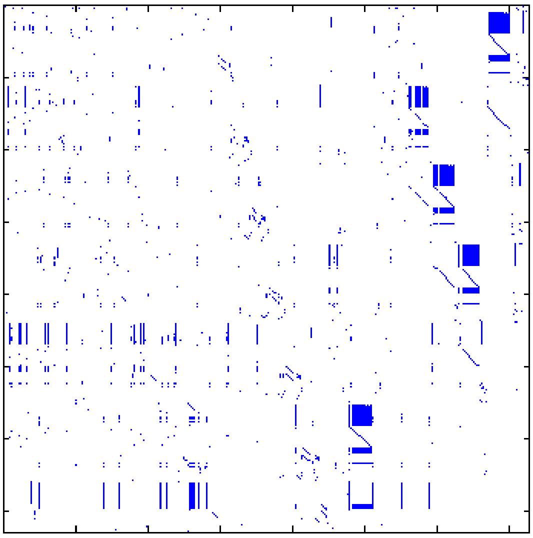


- Matrix $B \in \mathbb{R}^{363\times 363}$
- 3279 nonzeros
- Density 2.5%

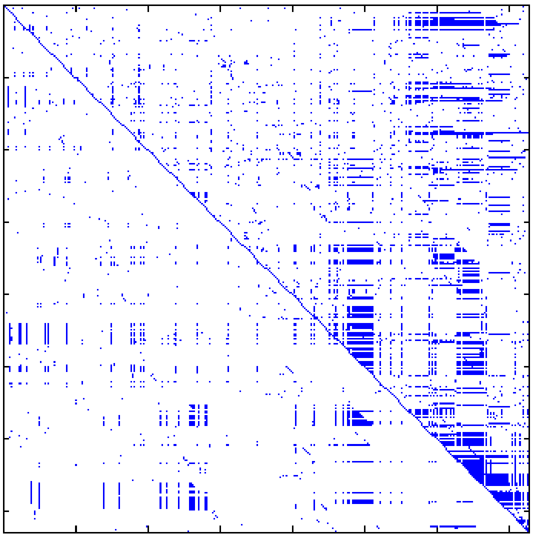


- Matrix B^{-1}
- 80284 nonzeros
- Density 61%
- Fill factor $\dfrac{80284}{3279} = 24.5$

What is required is a *representation* of B^{-1}, also referred to as an **invertible representation** of B. Even when the LU decomposition of B is computed using standard Gaussian elimination, the resulting representation of B^{-1} is usually very much more efficient with respect to sparsity.

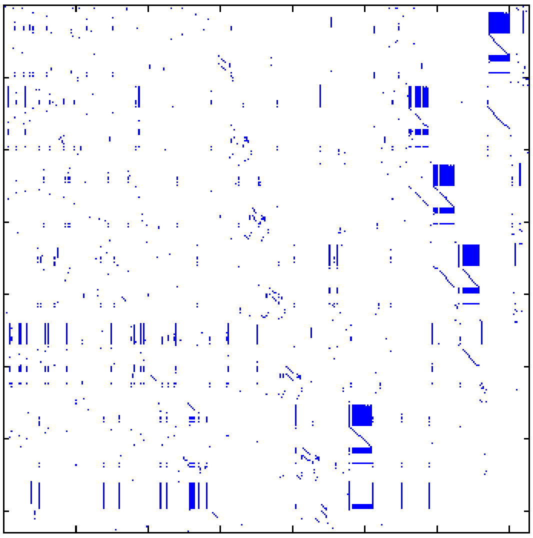


- Matrix $B \in \mathbb{R}^{363\times 363}$
- 3279 nonzeros
- Density 2.5%

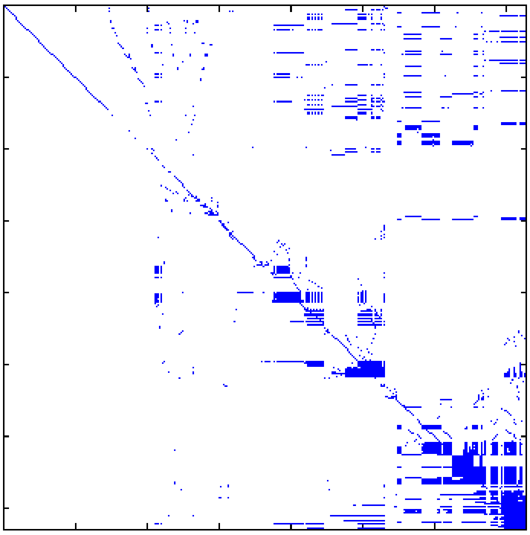


- Matrix $\left[\begin{smallmatrix} & U \\ L & \end{smallmatrix}\right]$
- 7850 nonzeros
- Density 6.0%
- Fill factor $\dfrac{7850}{3279} = 2.4$

When Gaussian elimination chooses pivots to preserve sparsity, efficiency is further improved.



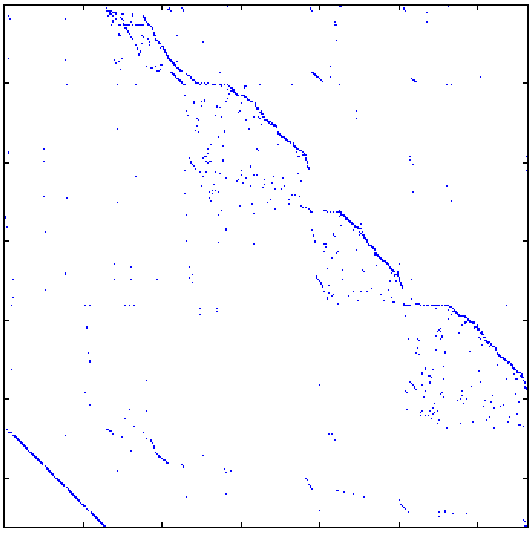
- Matrix $B \in \mathbb{R}^{363\times 363}$
- 3279 nonzeros
- Density 2.5%



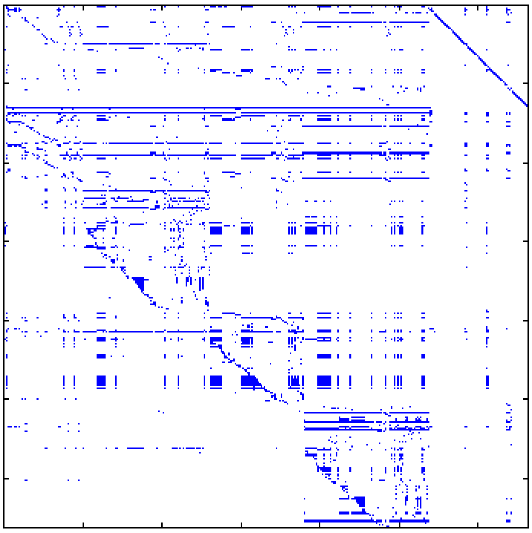
- Matrix $\left[\begin{smallmatrix} & U \\ L & \end{smallmatrix}\right]$
- 5043 nonzeros
- Density 3.8%
- Fill factor $\dfrac{5043}{3279} = 1.5$

Exploiting hyper-sparsity in the revised simplex method

For important classes of LP problems, further efficiency may be obtained by exploiting **hyper-sparsity**. Such LP problems have the property that B^{-1} is not dense.



- Matrix $B \in \mathbb{R}^{663\times 663}$
- 1712 nonzeros
- Density 0.4%
- LP test problem NESM



- Matrix B^{-1}
- 10765 nonzeros
- Density 2.4%
- Fill factor $\dfrac{80284}{1712} = 6.3$

If B^{-1} is sparse then $\boldsymbol{\pi}_p = B^{-T}\boldsymbol{e}_p$ is sparse and $\widehat{\boldsymbol{a}}_q = B^{-1}\boldsymbol{a}_q$ is typically sparse. LP problems for which B^{-1} is sparse are said to be **hyper-sparse**. However, B^{-1} is still not formed explicitly: it is more efficient to form $B = LU$ using sparse GE. Indeed, for problems when B^{-1} is sparse the LU decomposition can formed (almost) entirely by permutation rather than numerical elimination.

To demonstrate how hyper-sparsity may be exploited when solving systems involving B requires a more explicit representation of the LU decomposition of B. Consider L and U columnwise so that, in general, each column consists of a pivot value η_i and a vector $\boldsymbol{\eta}_i$ with zero i^{th} entry. Triangular substitution may then be applied to $\boldsymbol{b}$ as in (5).

```
For $i = 1, \ldots, N$
    $b_i := \dfrac{b_i}{\eta_i}$
    $\boldsymbol{b} := \boldsymbol{b} - b_i\boldsymbol{\eta}_i$                    (5)
End
```

If $B^{-1}\boldsymbol{b}$ is sparse then, in (5), b_i is zero in most cases so most multiplications and additions are with zero. Efficiency is improved by avoiding the operations with η_i and $\boldsymbol{\eta}_i$ if b_i is zero as in (6). However, for hyper-sparse LP problems the cost of testing $b_i \neq 0$ can dominate! This motivates more sophisticated techniques to identify the few columns of L and U for which non-trivial operations are needed.

```
For $i = 1, \ldots, N$
    If $b_i \neq 0$ then
        $b_i := \dfrac{b_i}{\eta_i}$
        $\boldsymbol{b} := \boldsymbol{b} - b_i\boldsymbol{\eta}_i$                (6)
    End
End
```

H and McKinnon (1998–2005), Bixby (1999)
Clp, Koberstein and Suhl (2005–2008)

Exploiting sparsity in interior point methods

When interior point methods (IPM) are applied to a general LP problem (4) the principal computational requirement is to solve systems of **normal equations** of the form

$$(A\Theta A^T)\boldsymbol{x} = \boldsymbol{b}, \tag{7}$$

where Θ is a diagonal matrix with positive entries. The matrix $G = A\Theta A^T$ is symmetric and positive definite and several such systems may have to be solved in each IPM iteration. Traditionally a **Cholesky decomposition** $G = LL^T$ is formed, where L is a general lower triangular matrix, so $G\boldsymbol{x} = \boldsymbol{b}$ may be solved as $L\boldsymbol{y} = \boldsymbol{b}$ by forward substitution and then $L^T\boldsymbol{x} = \boldsymbol{y}$ by backward substitution. Since Θ changes each IPM iteration, much of the decomposition process must be repeated since updates are not possible.

For a sparse LP problem, although A is sparse, entries of G are given by inner products of pairs of rows of A (weighted by Θ) so G is likely to be more dense than A. Although special sparsity-exploiting methods are used to form the Cholesky decomposition, further fill-in can be expected. For large LP problems the memory required to form L (and the computation required) may be prohibitive. For some such problems it may be reasonable to use the simplex method since the LU decomposition of the basis matrices incurs relatively little fill-in.

For very large scale sparse LP problems there has been recent work on the use of **matrix-free** methods for IMP [2]. These use **iterative methods** to solve $G\boldsymbol{x} = \boldsymbol{b}$ and are generally based on the **conjugate gradient method** (CG), motivated by the fact that it requires only matrix-vector products with G rather than any decomposition which would be vulnerable to fill-in. Since the product $G\boldsymbol{z}$ is formed as $A\Theta A^T\boldsymbol{z}$ via three products with A^T, Θ and A, it reduces the computation to operations on original sparse data. With CG, an approximate solution of $G\boldsymbol{x} = \boldsymbol{b}$ may be obtained in a small number of iterations if the eigenvalues of G lie in a corresponding number of clusters. Unfortunately it is very rare for this to occur as a natural consequence of the class of LP being solved so it is generally necessary to **precondition** the system $G\boldsymbol{x} = \boldsymbol{b}$ using a matrix P as $PGP^{-1}P\boldsymbol{x} = P\boldsymbol{b}$ so that the matrix $\widetilde{G} = PGP^{-1}$ has the desired spectral property. To be able to form $\widetilde{G}\boldsymbol{z}$ as $PGP^{-1}\boldsymbol{z}$ it is clearly essential that the cost of forming P and operating with it and its inverse does not compromise the potential gains of using CG. In most matrix-free IPM the preconditioner P is some approximation to the Cholesky matrix L so that $PGP^{-1} \approx I$. In the classes of problems solved successfully via matrix-free IPM the preconditioners contain only a very few columns of L, completed with identity columns.

Exploiting structure in convex optimization

Problem structure in convex optimization is generally manifested in the constraint matrix. There are many classes of structure and specialised methods have been developed to exploit this. The two fundamental structures are **row-linked block-angular form** and **column-linked block-angular form**. For row-linked block-angular problems the constraint matrix has the following form.

$$A = \begin{bmatrix} A_{00} & A_{01} & A_{02} & \dots & A_{0N} \\ & A_{11} & & & \\ & & A_{22} & & \\ & & & \ddots & \\ & & & & A_{NN} \end{bmatrix}$$

The constraints corresponding to the submatrix A_{00} involve all the decision variables in the problem and are often referred to as **linking constraints**. If the problem to be solved is an LP these constraints link what would, otherwise, be N independent LP problems. For this reason, problems with a row-linked block-angular constraint matrix are commonly referred to as **decentralised planning problems**, where local plans are made with respect to constraints on shared resources. The term **Dantzig-Wolfe** is also used to refer to this structure since it relates to an associated specialist solution procedure.
For column-linked block-angular problems the constraint matrix has the following form.

$$A = \begin{bmatrix} A_{00} & & & & \\ A_{10} & A_{11} & & & \\ A_{20} & & A_{22} & & \\ \vdots & & & \ddots & \\ A_{M0} & & & & A_{MM} \end{bmatrix}$$

The variables corresponding to the submatrix A_{00} involve all the constraints in the problem and are often referred to as **linking variables**. If the problem to be solved is an LP these variables link what would, otherwise, be M independent LP problems. In the context of stochastic LP these are variables whose values are chosen with respect to M **scenarios** whose variables can be chosen independently for a particular assignment of linking variables. The term **Benders** is also used to refer to this structure since it relates to an associated specialist solution procedure.

References

[1] R. E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling. MIP: Theory and practice closing the gap. In M. J. D. Powell and S. Scholtes, editors, *System Modelling and Optimization: Methods, Theory and Applications*, pages 19–49. Kluwer, The Netherlands, 2000.

[2] J. Gondzio. Matrix-free interior point method. *Computational Optimization and Applications*, 51:457–480, 2012.

[3] J. A. J. Hall and K. I. M. McKinnon. Hyper-sparsity in the revised simplex method and how to exploit it. *Computational Optimization and Applications*, 32(3):259–283, December 2005.

[4] A. Koberstein. Progress in the dual simplex algorithm for solving large scale LP problems: techniques for a fast and stable implementation. *Computational Optimization and Applications*, 41(2):185–204, November 2008.